Exascale Nearby Storage

Larry Kaplan (lkaplan@cray.com)

Dave Henseler (dah@cray.com)

Cray, Inc.

Exascale systems will require I/O bandwidth proportional to their computational capacity. Traditional globally shared file systems have limitations when used with exascale size systems:

- bandwidth does not scale economically to exascale size
- I/O traffic on the high speed network can impact and be impacted by other unrelated jobs
- I/O traffic on the storage server can impact and be impacted by other unrelated jobs

Description of research

One approach to avoiding those limitations is to configure multiple instances of smaller capacity, higher bandwidth storage closer to the compute nodes (nearby storage). The multiple instances can provide provide exascale size bandwidth and capacity in aggregate and can avoid much of the impact on other jobs. This approach does not provide the same file system semantics as a globally shared file system, in particular it does not provide file cache coherency or distributed locking, but there are many use cases where those semantics are not required. Other globally shared file system semantics are required, such as a consistent file name space, and must be provided by a nearby storage infrastructure.

In cases where the usage or lifetime of the application data is constrained, a globally shared file system provides more functionality than the application requires while at the same time limiting the bandwidth that the application could use if it were available. Nearby storage as described above reverses that, providing more bandwidth but not providing globally shared file system behavior. Use cases include checkpoint/restart files, application input files, staging data to and from globally shared file system, application temp/scratch space and kernel managed swap space.

Challenges addressed

Scalable bandwidth: By placing higher bandwidth storage close to the computation using it, more aggregate bandwidth can be provided than with an external globally shared file system.

Avoiding interference: By placing the storage close to the compute nodes, the I/O traffic can avoid interference with unrelated jobs.

Integrated software stack: This research would enable the creation of a unified software stack encompassing access to globally shared file systems, scalable staging between a globally shared file system and nearby storage, scalable nearby storage, I/O forwarding (ideally Posix compliant), a unified file system namespace, replication, fault tolerance and integration with the native high speed network infrastructure. The software stack will include both user space (for example staging) and kernel space (for example I/O forwarding and high speed network integration) components.

Maturity

Using high bandwidth storage (for example SSDs) to improve application I/O performance is becoming more common. HDFS and Hadoop have shown the benefits of nearby storage.

Uniqueness

Globally shared file systems are sufficient for most existing HPC systems, but the bandwidth demands of exascale systems require a more scalable (in both bandwidth and cost) solution.

Novelty

Existing infrastructure products, such as IOFSL, HDFS and PLFS, each address part of the larger problem, and may be part of an integrated software stack. An integrated stack as described would be scalable and optimized for exascale size systems.

Applicability

This approach will scale both up and down and so can provide more cost effective bandwidth for smaller size systems as well. Many different types of hardware can be used for nearby storage, including various forms of SSD as well as direct attach disk.

Effort

2 to 3 man-years to define the architecture. This will include prototyping and may include some parts of an initial reference implementation.

References

- HDFS: http://hadoop.apache.org/hdfs/docs/r0.22.0/hdfs_design.html
- PLFS: http://institutes.lanl.gov/plfs/
- IOFSL: http://www.mcs.anl.gov/research/projects/iofsl/
- Cray DVS: http://docs.cray.com/books/S-0005-4003/S-0005-4003.pdf
- IDC HPC End-User Study of the Evolution of Storage in Technical Computing: May 2011, IDC #228142, Volume: 1 High-Performance Computing End-User Study 2010: Special Study
- The NoLoSS Project: Investigating the roles of node local storage in Exascale systems: http://exascaleresearch.labworks.org/uploads/dataforms/A_OPH_ANL_NoLoSS_110214.pdf